

SYSTEM AND METHOD FOR ENCODING DSL INFORMATION STREAMS HAVING DIFFERING LATENCIES

CROSS-REFERENCE TO RELATED APPLICATIONS

5 This application claims the benefit of U.S. provisional patent application serial
60/206,068, filed on May 22, 2000, and entitled "Block Product Turbo BCH Codes and
QAM Soft Demapper for DSL DMT," which is hereby incorporated by reference in its
entirety. This application also claims the benefit of U.S. provisional patent application
serial 60/226,114, filed on August 18, 2000, and entitled "Multiple Latencies Multilevel
10 Turbo Coding for DSL," which is hereby incorporated by reference in its entirety.

BACKGROUND OF THE INVENTION

Field of the Invention

15 The present invention generally relates to communication systems, and more
particularly to a system and method for encoding DSL information streams having
differing latencies.

Discussion of the Related Art

20 In recent years telecommunication systems have expanded from traditional POTS
communications to include high-speed data communications as well. As is known, POTS
communications include not only the transmission of voice information, but also PSTN
(public switched telephone network) modem information, control signals, and other
information that is transmitted in the POTS bandwidth, which extends from
approximately 300 hertz to approximately 3.4 kilohertz.

New, high-speed data communications provided over digital subscriber lines (DSL), such as Asymmetric Digital Subscriber Line (ADSL), Rate Adaptive Digital Subscriber Line (RADSL), High-Speed Digital Subscriber Line (HDSL), *etc.* (more broadly denoted as xDSL) provide for high-speed data transmissions, as is commonly
5 used in communicating over the Internet. As is known, the bandwidth for xDSL transmissions is generally defined by a lower cutoff frequency of approximately 30 kilohertz, and a higher cutoff frequency which varies depending upon the particular technology. Since the POTS and xDSL signals are defined by isolated frequency bands, both signals may be transmitted over the same two-wire loop.

10 Indeed, twisted pair public telephone lines are increasingly being used to carry relatively high-speed signals instead of, or in addition to, telephone signals. Examples of such signals are ADSL (asymmetric digital subscriber line), HDSL (High Density Subscriber Line, T1 (1.544 Mb/s), and ISDN signals. There is a growing demand for increasing use of telephone lines for high speed remote access to computer networks, and
15 there have been various proposals to address this demand, including using voice over data systems to communicate signals via telephone lines at frequencies above the voice-band.

As is known, different applications often demand (or at least lend themselves to) different latency requirements. For example, applications of pure data transfer are often not sensitive to latency delays, while real-time voice communications are sensitive to
20 latency delays. As is also known, to accommodate maximum flexibility for providers and end users of ADSL services, forward error correction (FEC) may be selectively applied to the composite data streams to, or from, the central office ADSL modem. This

permits FEC to be included or excluded on a data service by data service basis within the composite data stream.

As an example of the mixed requirements for FEC in an ADSL service, consider transmitting a one-way data stream from the central office to a remote unit. The end user may require high reliability on the one-way channel because the channel may contain highly compressed digital data with no possibility for requesting retransmission. For this application, FEC is highly desired. On the other hand, voice services and duplex data services with their own embedded protocols may require minimum latency. As noted above, in real-time voice communication applications, latency delays are undesirable, while small transmission error may be tolerated (manifested as noise, which can be effectively filtered by the listener). Thus, in such an application, FEC may be optional.

FEC involves the addition of redundant information to the data to be transferred. The data to be transferred, along with the redundant data when added together, form what are commonly known as codewords. FEC in ADSL employs Reed-Solomon codes based on symbols of 8 bits to a byte. FEC in ADSL is rate adaptable, providing for various interleave depths and codeword lengths to support a range of data rates while maintaining constant interleave latency. An enhancement to FEC involves shuffling or interleaving the encoded data prior to transmission, then unshuffling or deinterleaving the data received at the remote DSL modem. Interleaving ensures that bursts of error noise during data transmission do not adversely affect any individual codeword in the transmission. If noise affects a particular frame of data, only a minimum number of

bytes of any particular codeword will be affected as the individual codewords are distributed across multiple frames.

The combination of Reed-Solomon encoding with data interleaving is highly effective at correcting errors caused by impulse noise in the service subscriber's local loop. In convolutional interleaving, after writing a byte into interleave memory, a previously written byte is typically read from the same memory.

Standard T1.413, Interface between Networks and Customer Installation – ADSL Metallic Interface provides for convolutional interleaving / deinterleaving along with Reed-Solomon coding as part of forward error correction (FEC). The standard provides an effective method for dealing with burst error channels in modern telecommunication systems. In DMT systems, two latency channels are supported: interleave data and fast data (without interleaving). Convolutional interleaving / deinterleaving is typically implemented by processing the Reed-Solomon encoded digital data sequence through a linear finite state shift register. In high bit rate applications like DMT, a random access memory (RAM) device may be used as the data storage means. Convolutional interleaving / deinterleaving is computation intensive. In software approaches that use a single address pointer and several modulo and addition operations to update the address pointer, system level concurrency and performance is adversely affected. Conversely, hardware approaches that utilize multiple pointers for interleaving / deinterleaving operations increase the complexity of the overall DSL system. The system performance trade-off introduced by FEC in the form of Reed-Solomon coding and convolutional interleaving can be described as increased data transmission reliability at the expense of increased channel latency.

U.S. Patent number 5,764,649 to Tong discloses a system and method compliant with the T1.413 standard. As illustrated in the '649 patent, both a "fast path" and an interleave path are provided downstream of the FEC. As taught in the '649 patent, two frames are output from multiplexer every frame period. One frame is sent through ADSL transmitter along a "fast path" while the other frame is sent along an "interleave path."

The fast path is so called simply because the data does not undergo the additional processing of interleaving, and therefore does not experience the additional delay imposed by de-interleaving at the receiving end of the communication system. However, all data from the incoming bit stream is passed through the FEC, and therefore encounters the latency delay associated therewith.

Accordingly, there is a need to provide an improved system and method for encoding DSL information streams to further minimize latency delays. Further, there is a desire to provide an improved system and method for encoding DSL information streams having differing latencies.

SUMMARY OF THE INVENTION

Certain objects, advantages and novel features of the invention will be set forth in part in the description that follows and in part will become apparent to those skilled in the art upon examination of the following or may be learned with the practice of the invention. The objects and advantages of the invention may be realized and obtained by means of the instrumentalities and combinations particularly pointed out in the appended claims.

5 To achieve the advantages and novel features, the present invention is generally directed to a system and method for encoding a DSL information bit stream and decoding a corresponding encoded DSL symbol. In accordance with one embodiment, an apparatus for encoding a DSL information bit stream is provided having a switch with an input configured to receive a DSL information bit stream and at least two outputs. An encoder is provided and coupled to a first output of the switch. A serial to parallel converter is provided and coupled to both an output of the encoder and a second output of the switch. Finally, a mapper is provided and coupled to an output of the serial to parallel converter through multiple paths. Preferably, a first coupling path between the serial to parallel converter and the mapper is a direct path and a second coupling path includes a second encoder.

10 In accordance with another embodiment, an apparatus is provided for decoding an encoded DSL symbol. The apparatus includes a soft demapper configured to generate multiple outputs, including an uncoded bit stream having an input configured to receive a DSL information bit stream and at least one coded bit stream. The apparatus further includes at least one decoder configured to decode the at least one coded bit stream. Finally, the apparatus includes a circuit configured to perform a hard demapping of both the uncoded bit stream and an output of the at least one decoder.

20 In accordance with yet another embodiment of the invention, a method is provided for encoding a DSL information bit stream. Preferably, the method operates by providing at least two paths between an input configured to receive the DSL information bit stream and a serial to parallel converter. The method providing a first encoder in one

of the at least two paths, and switches the DSL information bit stream through one of the at least two paths based upon a latency in the DSL information bit stream.

DESCRIPTION OF THE DRAWINGS

5 The accompanying drawings incorporated in and forming a part of the specification, illustrate several aspects of the present invention, and together with the description serve to explain the principles of the invention. In the drawings:

FIG. 1 is a diagram illustrating fundamental components in a transmitter constructed in accordance with one embodiment of the invention;

10 FIG. 2 is a diagram illustrating fundamental components in a receiver constructed in accordance with one embodiment of the invention;

FIG. 3 is a block diagram similar to FIG. 1, but illustrating fundamental components in a transmitter constructed in accordance with another embodiment of the invention;

15 FIG. 4 is a block diagram illustrating a serial concatenated turbo encoder;

FIG. 5 is a block diagram illustrating a parallel concatenated turbo encoder;

FIG. 6 is a diagram illustrating a product block code encoding scheme;

FIG. 7 is a block diagram illustrating an iterative turbo decoding scheme;

FIG. 8 is a diagram illustrating a product block coding scheme;

20 FIG. 9 is a diagram illustrating extrinsic information generated by the Chase algorithm; and

FIG. 10 is a diagram illustrating an 8 PAM with three bits and Gray labeling.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Having summarized various aspects of the present invention, reference will now be made in detail to the description of the invention as illustrated in the drawings. While the invention will be described in connection with these drawings, there is no intent to
5 limit it to the embodiment or embodiments disclosed therein. On the contrary, the intent is to cover all alternatives, modifications and equivalents included within the spirit and scope of the invention as defined by the appended claims.

Reference is now made to FIG. 1, which is a block diagram showing an apparatus 100 for encoding a DSL bit stream 102 in accordance with one embodiment of the
10 present invention. In this regard, a fundamental aspect of the preferred embodiment of the invention relates to the ability to vary the level or depth of encoding of a DSL bit stream, based upon the desired latency in communicating that bit stream. As previously described, certain applications like data transfer may not be sensitive to latency delays, and therefore a higher degree of encoding is often desired. Other applications, like voice
15 communications, desire minimal latency delays, while tolerating certain levels of transmission error. Therefore, in accordance with a fundamental aspect of the preferred embodiment, a mechanism is provided for switching a DSL bit stream 102 between a low latency path and a higher latency path within the apparatus 100.

In this regard, a switching mechanism 110 is provided. In the diagram of FIG. 1,
20 the switching mechanism 110 is illustrated as a simple two-position switch. It should be appreciated that this depiction has been presented purely for illustration purposes and that, in the preferred embodiment, electronic switching elements (*e.g.*, solid state switching elements) may be desired. In a first position (the position illustrated in FIG. 1),

the switching mechanism 110 connects the DSL bit stream 102 through a high latency path, which includes a forward error-correction encoder (FEC), such as a Reed-Solomon encoder 112 and an interleaver 114. It will be appreciated that other types of FEC may be implemented in the high latency path. For purposes of the preferred embodiment, an
5 encoder is provided in this high latency path to provide an additional measure of error correction capability, so that the DSL bit stream may be encoded and transmitted to a receiver with greater immunity to channel noise or other interference.

In a second position, the switching mechanism 110 may connect the DSL bit stream 102 directly to a serial to parallel converter 116. In the illustrated embodiment, an
10 additional encoder 118 (an outer block code) may be included to provide a smaller, but additional measure of error correction to the transmitted symbol.

Of course, a mechanism 120 is also provided to control the operation of the switching mechanism 110. The switch control mechanism 120 controls the position of the switching mechanism 110 in accordance with the latency demands of the DSL bit
15 stream 102. In this regard, the switch control mechanism 120 is provided with some indication as to the type of information being conveyed in the DSL bit stream 102. This information may be ascertained by the switch control mechanism 120 evaluating the signals on the DSL bit stream 102 themselves. Alternatively, this information may be provided through network layer transport mechanisms communicating to the switch
20 control mechanism 120. For purposes of the present invention, the implementation of the switch control mechanism is not deemed to be relevant, as the particular implementation of this device or feature is not considered to be a limitation upon the present invention.

As illustrated, the output of the interleaver 114 is also directed to the serial to parallel converter 116. The parallel data that is output from the serial to parallel converter 116 is then directed to a mapper 130 (also referred to as a symbol selector). In a manner that is known and understood by persons skilled in the art, the mapper 130 operates to select or generate an output symbol (*e.g.*, constellation point). Although various encoders may be utilized consistent with the scope of the invention, in the preferred embodiment, the encoder 140 is implemented as a parallel concatenated encoder (or turbo coder), which includes encoders 142 and 144 and an interleaver 146. Concatenated codes are now well known and need not be described herein. As will be appreciated, additional levels or degrees of coding may be added consistent with the invention.

Accordingly, what is generally illustrated in FIG. 1 is a transmitter/encoder for receiving a DSL bit stream 102 and generating a series of output symbols at 150, which can have varying degrees of error correction coding (*i.e.*, redundancy), based upon the desired or acceptable latency of the information being communicated through the DSL bit stream 102. In this regard, a switching mechanism 110 is configured and controlled by a switch control mechanism 120 to direct DSL bit stream 102 through either a higher latency path (comprising additional levels of error correction coding, such as FEC) or a lower latency path, which implements a lower degree of error correction coding.

Reference is made now to FIG. 2, which is a block diagram illustrating certain fundamental components implemented in a receiver constructed in accordance with a preferred embodiment of the invention. The illustrated input 202 is taken at a point in a receiver after equalization has been performed in a single carrier modulation. In

multicarrier modulations (e.g., DMT DSL), the input 202 is taken at a point in the receiver after the equalization and FFT and FEQ. The symbol received on the input 202 is then operated upon by a demapper 204. The demapper 204 operates to essentially perform the inverse of the mapper 130 illustrated in FIG. 1. As is known, due to noise and interference incurred in the communication channel, the received symbols generally do not exactly correspond to a constellation point. Instead, certain "decisions" must be made to determine the constellation point that is properly associated with the received symbol. As is well known, there are various mechanisms for implementing this type of decision making process. In the preferred embodiment, a "soft" demapper 204 is implemented. A more detailed discussion of the soft demapping process implemented in the preferred embodiment will be described in more detail below. For purposes of the broader concepts of the invention, however, any appropriate or conventional demapper may be utilized.

As illustrated, multiple outputs are generated from the soft demapper 204. The decoder 210 may be configured to perform the inverse of the operation of the encoder 140 of FIG. 1. In the preferred embodiment, the decoder 210 includes decoders 212 and 214. The output from the decoder 210, as well as the uncoded bits 206, are directed to a hard demapping circuit 220. As will be appreciated by persons skilled in the art, since the modulation is a coded modulation, a soft demapper is first implemented before performing the hard demapping.

The circuit 220 also operates to determine whether the transmitted signal was generated through the low latency path or high latency path illustrated in FIG. 1. If the symbol was generated through the low latency path, then the circuit 220 operates to

assign the output to a first output path 230. As illustrated, the signal output path 230 may be processed by a low latency outer decoder 232, if the encoder 118 was provided and implemented in FIG. 1. Alternatively, the circuit 220 outputs the signal onto path 240, which is passed through an interleaver 242 and Reed-Solomon decoder 244. As should
5 be understood, the deinterleaver 242 performs the inverse operation of interleaver 114, while the Reed-Solomon decoder 244 performs the inverse operation to the Reed-Solomon encoder 112.

Reference is now made briefly to FIG. 3, which is a block diagram similar to the diagram of FIG. 1, but illustrating other concepts of one implementation of the present
10 invention. Specifically, in accordance with the broader concepts and teachings of the present invention, it should be understood that the DSL bit stream may be processed in accordance with more than two degrees or levels of latency. In this regard, a switch 310 may be provided to receive the DSL bit stream 302 and directed to any of a number of Reed-Solomon encoders, each Reed-Solomon encoder 312, 314, and 316 having a
15 differing degree of decoding and differing depths of interleaving. Consequently, the error correction capabilities among the various Reed-Solomon encoders 312, 314, and 316 differ. Consequently, the latency associated with each of the Reed-Solomon encoders 312, 314, and 316 differ correspondingly. A direct path 318 between the switch 310 and serial to parallel converter 320 may also be provided. The remainder of the diagram of
20 FIG. 3 may be implemented in the same fashion as that described in connection with FIG. 1.

Although not specifically shown, the implementation of a receiver for receiving a symbol generated by the circuitry of FIG. 3, could be implemented similarly to the

receiver of FIG. 2. However, instead of having a single deinterleaving 342 and Reed-Solomon decoder 244, multiple deinterleavers and multiple Reed-Solomon decoders would be provided.

In accordance with yet another embodiment of the invention, a method is provided for encoding a DSL information bit stream. As should be appreciated from the foregoing discussion, the method operates by providing at least two paths between an input configured to receive the DSL information bit stream and a serial to parallel converter. The method providing a first encoder in one of the at least two paths, and switches the DSL information bit stream through one of the at least two paths based upon a latency in the DSL information bit stream.

Having described the top-level architecture and operation of a system and method constructed in accordance with the invention, the discussion will now focus on implementation details of certain preferred embodiments of the invention. For example, FIGS. 2 and 4 summarily described the use of encoders (e.g., 142 and 144) in the transmitter, at the input of the mapper 130. Consistent with the invention, various types of encoders may be implemented. In the preferred embodiment, however, concatenated codes (sometimes referred to as turbo codes) are implemented.

As is known, turbo codes cover channel coding techniques that combine at least 2 "light codes" at the transceiver side, separated by a decoupling device such as an interleaver, and jointly soft decode them iteratively at the receiver side, in order to benefit from the performance of the equivalent "longer code" at a reasonable price. In this regard, the "turbo" effect takes place at the receiver side, insofar as performance is increased since the 2 soft input soft output SISO decoders (See FIG. 7) dedicated to each

code exchange soft information that describes the reliability of the hard decision that could have been made if hard decoding had been used. It should be appreciated that the same information has been coded twice. When the reliability is high enough, a hard decision is finally made. Therefore, the term "turbo codes" may be somewhat misleading, since truly the codes used are "concatenated," and the decoder is, as a matter of fact, a soft turbo-decoder. In the name turbo code, "turbo" thus refers to the fashion of decoding. In addition, the soft information that propagates from one SISO decoder to the other one is generally called "extrinsic information."

The two generally two categories or families of turbo codes: convolutional turbo codes and block turbo codes. As is known, convolutional turbo codes combine two convolutive codes C_b and C_a either in a serial (FIG. 4) or parallel (FIG. 5) fashion. As mentioned in the turbo code definition, to be full a "turbo code scheme" requires a SISO soft input soft output decoder for each code. Most SISO decoders involved in convolutive turbo codes (either parallel or serial) make use of Soft Output Viterbi Algorithm or simplified versions of the Optimum MAP BCJR algorithm. Since these methodologies and implementations are known by persons skilled in the art, they need not be further described herein.

In the preferred embodiment of the invention, the encoders are implemented as product block turbo codes. As is known, product block codes have long been known. However, the iterative hard decoding for such codes is to be tedious and yields bad performance. Product block turbo codes lift the intrinsic "hard iterative decoding" limitation by introducing a "soft iterative procedure". Product Block codes put the information bits in a matrix fashion (see FIG. 6). A first block code C_a encodes the rows,

a second block code C_b encodes the columns (*including the parity columns*). More features of Product Block Codes are discussed below.

The SISO block decoder (FIG. 7) involves in product block codes is based on what is referred to as the CHASE algorithm, which will be further detailed below.

5 One of the classical and difficult issue in channel coding concerns the choice of an efficient code. For Turbo codes, the issue remains. Based on experimental results, it have been found that convolutive turbo codes, either serial or parallel, exhibit good performance for low power & low spectral efficiency applications such as deep space probe. In contrast, product block codes target high spectral efficiency projects. Practically,
10 for code rates lower than 0.7, convolutive codes can be used, if the rate goes above this threshold up to 0.9 product block turbo codes are preferred.

Most of the turbo decoders follow the paradigm illustrated in FIG. 7. Raw data are processed first by the Soft Input Soft Output SISO decoder 402 devoted to code a. The relevant soft output, very often called extrinsic information a, is interleaved by interleaver
15 404 and then processed by the SISO decoder 406. The soft output b is deinterleaved by deinterleaver 408 to feed the next iteration of the turbo decoder.

The soft output or the extrinsic information measures the reliability of the hard decision that could have been made if a hard decoder were used. Therefore, as soon as the process goes through sufficient iterations, the extrinsic information changes according to
20 the improvement of the hard decision that could have been made if a hard decoder were used. When the extrinsic information reaches a steady state, a high reliable “hard decision” is performed to generate a hard output 410. Generally, a few (three to five) iterations are sufficient to get very close to the Shannon bound.

Convolutional SISO algorithms generally rely either on “soft Viterbi algorithm SOVA” or on the known BCJR (Bahl, Cocke, Jelinek, Raviv) algorithm. Both of them model the convolutional code as a finite state Markov Chain. As is known, the Viterbi
5 algorithm estimates the whole state path of the Markov Chain. Although based on a MAP criterion, the Viterbi algorithm is suboptimal, since a certain number of transitions are dropped from time to time to reduce complexity. As is further known, the BCJR algorithm searches for the current and the future states of the Markov Chain at any time according to a MAP criterion. The complexity of SISO schemes devoted to
10 convolutional codes is generally quite high. Conversely, the SISO algorithms used for product block codes are simpler and are based on the Chase algorithm.

General properties of Product Codes

As illustrated in FIG. 8, product block codes put the information bits into matrix
15 fashion. The k_b rows are encoded by a first code $C[n_a, k_a, d_a]$, where d_a is the minimum Hamming distance, k_a is the number of information bits, and n_a is the code length. The n_a columns are encoded by a second code $C[n_b, k_b, d_b]$. Therefore all the n_b rows are code words of C_a , and all the n_a columns are codewords of C_b . The parameters of the product code $C[n, k, d]$ are the product of the elementary codes parameters:

$$20 \quad n = n_a \times n_b, \quad k = k_a \times k_b, \quad d = d_a \times d_b \quad \text{Equation 1}$$

Other interesting properties of product block codes are related to the “code spectrum”, i.e. the number $A(w)$ of codes words with a certain weight w (number of bits equal to one). The number of code words (*of the Product Block Code*) with the minimum weight is the product of the number of code words with the minimum weight of the two original codes is given by:

$$A(d_{\min}) = A_a(d_{\min,a})A_b(d_{\min,b}) \quad \text{Equation 2}$$

Soft decoding of Block Codes: CHASE Algorithm

Complexity of the Maximum likelihood approach

Although maximum likelihood decoding is optimum, it generally implies an unrealistic complexity, when the unknown takes discrete values and when their number is higher than 10. Assume, for example, that a row \mathbf{r} (n_a components) of the matrix is transmitted. The associated received vector \mathbf{x} contains additive noise:

$$\mathbf{x} = \mathbf{r} + \mathbf{n} \quad \text{Equation 3}$$

If the noise is white and gaussian, the maximum likelihood approach looks for the code word, which minimizes the following Euclidian distance is:

$$\mathbf{r}^{opt} = ARG\{Min_{1 \leq i \leq 2^{k_a}} \{\|\mathbf{x} - \mathbf{r}^i\|^2\}\} \quad \text{Equation 4}$$

This requires a search amongst 2^{k_a} discrete binary vectors, resulting in an exponentially increasing complexity. In equation 4, \mathbf{r}^i designates the “code word i ”, of code C_a :

$$\mathbf{r}^i = [r_1^i \dots r_m^i \dots r_{n_a}^i] \quad \text{Equation 5}$$

Each component takes binary values 0, or 1.

CHASE proposed an algorithm which approximates the maximum likelihood sequence decoding of block codes with a low computation complexity and a small performance degradation.

CHASE algorithm

5 Instead of reviewing all the codes words of C_a , the CHASE algorithm searches for these located in a subspace of the code. If a BPSK modulation is used, the received vector, still associated to a row \mathbf{r} (for simplicity's sake) is:

$$\mathbf{x} = \tilde{\mathbf{r}} + \mathbf{n} \quad \text{Equation 6}$$

10 where:

$$\begin{aligned} \tilde{r}_m &= 1, \text{ if } r_m = 1 \\ \tilde{r}_m &= -1, \text{ if } r_m = 0 \end{aligned} \quad \text{Equation 7}$$

We denote \mathbf{y} , the binary vector deduced after hard slicing of \mathbf{x} :

$$y_m = \frac{1}{2}(1 + \text{sgn}[x_m]) \quad \text{Equation 8}$$

The CHASE algorithm is based on 4 steps:

20 **Step 1.** Find the P least reliable bits of \mathbf{y} .

Step 2. Elaborate the 2^P associated “test sequences”, \mathbf{z}' , $1 \leq i \leq 2^P$.

Step 3. Hard decode the 2^P test sequences, store the results which spans the subspace .

25 **Step 4.** Decode \mathbf{x} based on maximum likelihood in $\tilde{\Omega}$, and compute the reliability of the decision to elaborate the soft output.

With regard to step 1, the P least reliable bits are associated to the P smallest values of $|x_m|$.

With regard to step 2, to built the test sequences, the most reliable bits are kept, and the P least reliable takes any binary values 0, or 1. This leads to 2^P possibilities.

5 With regard to step 4, the likelihood approach limited to leads to the decision \mathbf{d} :

$$\mathbf{d} = \underset{\tilde{\mathbf{r}} \in \tilde{\Omega}}{\text{ARG}\{\text{Min}\{\|\mathbf{x} - \tilde{\mathbf{r}}\|^2\}}}$$

Equation 9

where $\tilde{\Omega}$ is the extension of according to the BPSK modulation, see equation 7. The
10 reliability of decision \mathbf{d} is evaluated per bit m . The reliability of bit m is based on the following Log-likelihood ratio:

$$\Lambda(d_m) = \text{Log}\left\{\frac{\Pr[\tilde{r}_m = +1 / \mathbf{x}]}{\Pr[\tilde{r}_m = -1 / \mathbf{x}]}\right\}$$

Equation 10

After some calculation and manipulation, the Log-likelihood ratio becomes
15 approximately:

$$\Lambda(d_m) \approx \frac{1}{2\sigma^2} (\|\mathbf{x} - \tilde{\mathbf{r}}^{-1(m)}\|^2 - \|\mathbf{x} - \tilde{\mathbf{r}}^{+1(m)}\|^2)$$

Equation 11

where $\tilde{\mathbf{r}}^{-1(m)}$ is the closest code word to \mathbf{x} in \tilde{C}_a with bit m equals to -1 , $\tilde{\mathbf{r}}^{+1(m)}$ is the closest code word to \mathbf{x} in \tilde{C}_a with bit m equals to $+1$. It is particularly noted that if:

$$\mathbf{d} = \tilde{\mathbf{r}}^{-1(m)}$$

Equation 12

the Log-likelihood ratio is negative. If:

$$\mathbf{d} = \tilde{\mathbf{r}}^{+1(m)}$$

Equation 13

the likelihood ratio if positive. Therefore, the Log-likelihood ratio of bit m , bears its sign.

The Log-likelihood ratio might be rewritten as:

$$\Lambda(d_m) \approx \frac{1}{2\sigma^2} \left(\|\mathbf{x} - \tilde{\mathbf{r}}^{-1(m)}\|^2 - \|\mathbf{x} - \tilde{\mathbf{r}}^{+1(m)}\|^2 \right) d_m \quad \text{Equation 14}$$

5 Based on a straightforward calculation, it is possible to show that if $\sigma^2 = 2$, then:

$$\Lambda(d_m) \approx x_m + w_m = x'_m \quad \text{Equation 15}$$

where w_m is the extrinsic information stemming from the likelihood evaluation. This

10 extrinsic information is added to the soft input to elaborate the soft output x'_m which reliability is contained in its absolute value: $|x'_m|$. When this absolute value is growing, so does the reliability.

Simplified soft output.

15

The Log-likelihood given by Equation 14 is the “soft output” of the decoding. It becomes an additive correction of the input data for a particular normalization. Nevertheless, the search for both code words $\tilde{\mathbf{r}}^{+1(m)}$ and $\tilde{\mathbf{r}}^{-1(m)}$ in the whole code \tilde{C}_a is still unrealistic. This optimum soft output may be replaced by the following simplified

20 version in the CHASE algorithm:

$$x'_m \approx \frac{1}{4} (\|\mathbf{x} - \mathbf{a}\|^2 - \|\mathbf{x} - \mathbf{d}\|^2) d_m \quad \text{Equation 16}$$

where \mathbf{d} is the likelihood decision in $\tilde{\Omega}$, d_m is the bit m of \mathbf{d} . \mathbf{a} is the “challenger” of \mathbf{d} . It is the closest (in $\tilde{\Omega}$ code word to \mathbf{x} , but with a bit m equal to $-d_m$. According to equation

25 (15), the extrinsic information w_m is the difference:

$$\frac{1}{4}(\|\mathbf{x} - \mathbf{a}\|^2 - \|\mathbf{x} - \mathbf{d}\|^2)d_m - x_m = w_m \quad \text{Equation 17}$$

If the challenger \mathbf{a} does not exist, than the soft output is chosen to be equal to in coherence with Equations 15 and 16:

$$x'_m = x_m + \beta d_m \quad \text{Equation 18}$$

As a summary, for BPSK modulation, the CHASE algorithm associates to a soft input \mathbf{x} a soft output \mathbf{x}' (according to either Equation 15 or 16). Their difference is the extrinsic information \mathbf{w} (see FIG. 9).

As a compact useful notation, consider:

$$\text{CHASE}[\mathbf{x}] = \mathbf{x}' \quad \text{Equation 19}$$

15 ***Iterative Soft Decoding of Product Block Codes: Block Product Turbo Codes for BPSK modulation***

Assuming that the received data consists in a matrix \mathbf{X} with n_a columns and n_b rows, according to the fashion described above. One iteration of the turbo decoding of \mathbf{X} is split into two “half iterations”: the soft decoding of all the rows on the one hand and all the columns on the other hand, based both on the CHASE algorithm.

The iterative scheme is as follows.

Iteration 0.

First Half Iteration 0. Rows. Soft decode all the rows of matrix \mathbf{X} . Half iteration 0 does not use any extrinsic information as input. Denoting $\mathbf{SI}_R[0,0]$ as the soft input at the half iteration 0 (the index R stands for rows, although it could be of course replaced by the same operation on the columns). The soft output $\mathbf{SO}_R[0,0]$ after the first half

iteration 0 may be obtained by soft decoding (based on the CHASE algorithm devoted to code C_a) of all the rows of \mathbf{X} . In summary,

$$\begin{aligned}\mathbf{SI}_R[0,0] &= \mathbf{X} \\ \mathbf{SO}_R[0,0] &= \text{CHASE}_R[\mathbf{X}] \\ \mathbf{W}_R[0,0] &= \mathbf{SO}_R[0,0] - \mathbf{SI}_R[0,0]\end{aligned}\tag{Equation 20}$$

5

Second Half iteration 0, Columns. The soft input of the second half iteration 0 $\mathbf{SI}_C[0,1]$ includes part of the extrinsic information stemming from the first half iteration 0:

$$\mathbf{SI}_C[0,1] = \mathbf{X} + \alpha[0,1]\mathbf{W}_R[0,0]\tag{Equation 21}$$

10

The soft output of the second half iteration $\mathbf{SO}_C[0,1]$ results from the soft decoding (based on the CHASE algorithm devoted to code C_b) of all the columns of the soft input given by Equation 21:

15

$$\mathbf{SO}_C[0,1] = \text{CHASE}_C[\mathbf{SI}_C[0,1]]\tag{Equation 22}$$

The relevant extrinsic information $\mathbf{W}_C[0,1]$ is still the difference between the soft output and the soft input:

$$\mathbf{W}_C[0,1] = \mathbf{SO}_C[0,1] - \mathbf{SI}_C[0,1]\tag{Equation 23}$$

20

At the first iteration, the extrinsic information is not highly reliable, the reliability increases with each iteration. The extrinsic information which feeds the next SISO block is thus weighted by a coefficient tuned according to the iteration. This coefficient is increasing from 0 to 1.

25

Iteration $m+1$. $m \geq 0$

First half iteration $m+1$. Rows.

Soft Input: $\mathbf{SI}_R[m+1,0] = \mathbf{X} + \alpha[m+1,0]\mathbf{W}_C[m,1]$
 Soft Output: $\mathbf{SO}_R[m+1,0] = \mathbf{CHASE}_R[\mathbf{SI}_R[m+1,0]]$
 Extrinsic Information: $\mathbf{W}_R[m+1,0] = \mathbf{SO}_R[m+1,0] - \mathbf{SI}_R[m+1,0]$.

5

Second half iteration m+1. Columns.

Soft Input: $\mathbf{SI}_C[m+1,1] = \mathbf{X} + \alpha[m+1,1]\mathbf{W}_R[m+1,0]$
 Soft Output: $\mathbf{SO}_C[m+1,1] = \mathbf{CHASE}_C[\mathbf{SI}_C[m+1,1]]$
 Extrinsic Information: $\mathbf{W}_C[m+1,1] = \mathbf{SO}_C[m+1,1] - \mathbf{SI}_C[m+1,1]$.

10

Generally 3 to 4 iterations are sufficient to lead to almost the maximum expected gain of turbo decoding.

15 **BLOCK PRODUCT TURBO CODES FOR ADSL DMT**

Extension to the ADSL-DMT constraints

The idea is to replace the actual coded QAM modulation, the interleaver and the Reed-Solomon code by both a soft demapper of QAM and a block product turbo BCH code. In what follows, we will refer to the “classical chain” as the chain without turbo codes. The matrix fashion inherent to block product behaves as a matter of fact as an interleaver. To introduce block product turbo codes requires the satisfaction of ADSL constraints.

20

Block Product Turbo BCH, BPTBCH, scheme suited to ADSL framing constraints

25

The Reed-Solomon leads to 255 bytes words. BCH codes are binary cyclic codes. BCH codes with 255 bits are thus convenient (*8 BCH words fit one RS word*). Starting from the fact that the rate of the product block code is the product of the original rates, to keep a reasonable rate, consider $(255,247)^2$ BPTBCH and $(255,247)(255,239)$ BPTBCH

with respective rates equal to 0.937 and 0.907. Both schemes are equivalent to an interleaver depth of at least 29 bytes.

SOFT DEMAPPING

5 The following presents a mathematical discussion (and illustration) of soft demapping. It should be appreciated that the following is presented purely for the purpose of illustration, and should not be viewed as limiting in any way upon the broader concepts of the present invention.

ML partitioning of the real axis

10 For simplicity, consider an 8 PAM, based on 3 bits, $b_2b_1b_0$ and a Gray labeling (see FIG 11). We introduce also the ML partition of the real axis in the presence of additive white gaussian noise. This means that the 8 classes $C_k, 0 \leq k \leq 7$ associated to each of the 8 three bits words are based on the middle of the segments defined by the values associated to each of the 8 three bits words (see FIG. 10). We call r the received
15 information. The target is to define a soft information for each of the three bits $b_2b_1b_0$, based on r . The “optimum” soft information makes use of the MAP or the Log(MAP) ratio:

$$L_i(r) = \frac{p(b_i = 1/r)}{p(b_i = 0/r)} = \frac{\sum_{\mathbf{b}_{-i}} p(\mathbf{b}_{-i}, b_i = 1/r)}{\sum_{\mathbf{b}_{-i}} p(\mathbf{b}_{-i}, b_i = 0/r)} \quad \text{Equation 24}$$

where \mathbf{b}_- is the 2 bits word built with the bits different from i . Instead of considering the $\text{Max}[\text{Log}(\text{MAP})]$ argument and the Bayes rule to simplify Equation 25, we introduce the simple idea of “tangent BPSK” modulation.

Soft bits based on “tangent BPSK” modulation

5 To explain the idea of “tangent BPSK” modulation, we consider an example and assume that $r = 1.1$ (see FIG. 10). Our procedure yields “soft bits” $\tilde{b}_i(r), 0 \leq i \leq 2$ based on r . First of all, a class membership is defined, straightforwardly C_5 in FIG. 10. Then, one “tangent BPSK” modulation is introduced per bit, conditionally to class C_5 .

10 a) **Soft bit** $\tilde{b}_0(r)$. In class C_5 , $b_2b_1b_0 = 111$. The present procedure states 110 as the “challenger” of 111 for bit b_0 . The relevant “tangent BPSK” modulation takes place between 111 and 110. The relevant values have thus to be tuned based on translation and scaling to lead to “BPSK like” ranges:

15
$$\tilde{b}_0(r) = r - 1 = 0.1 \quad \text{Equation 25}$$

b) **Soft bit** $\tilde{b}_1(r)$. In class C_5 , $b_2b_1b_0 = 111$. The present procedure states 101 as the “challenger” of 111 for bit b_1 . The relevant “tangent BPSK” modulation takes place between 111 and 101. The relevant values have thus to be tuned based on translation and

20 scaling to lead to “BPSK like” ranges:

$$\tilde{b}_1(r) = r - 2 = -0.9 \quad \text{Equation 26}$$

c) **Soft bit** $\tilde{b}_2(r)$. In class C_3 , $b_2b_1b_0 = 111$. The present procedure states 011 as the “challenger” of 111 for bit b_2 . The relevant “tangent BPSK” modulation takes place between 111 and 011. The relevant values have thus to be tuned based on translation and scaled to lead to “BPSK like” ranges:

$$5 \quad \tilde{b}_2(r) = \frac{r}{3} = 0.36 \quad \text{Equation 27}$$

Then, any SISO BPSK (+0.5, -0.5) based procedure may be processed.

10 The foregoing description has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Obvious modifications or variations are possible in light of the above teachings. The embodiment or embodiments discussed were chosen and described to provide the best illustration of the principles of the invention and its practical application

15 to thereby enable one of ordinary skill in the art to utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated. All such modifications and variations are within the scope of the invention as determined by the appended claims when interpreted in accordance with the breadth to which they are fairly and legally entitled.